

Practica 2.

Lisardo Gayán Tremps

José Luis Melo

2 June, 2019

Contents

1 - Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?	2
2 - Integración y selección de los datos de interés a analizar	2
3 - Limpieza de datos	5
3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos? .	5
3.2. Identificación y tratamiento de valores extremos.	8
4. Análisis de los datos.	9
4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar)	9
4.2. Coprobación de la normalidad y homogeneidad de la varianza.	9
4.3. Aplicación de pruebas estadísticas para comprar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.	9
5. Representación de los resultados a apartir de tablas y gráficas.	9
6. Resolución del problema. A partir de los resultados obtenidos. ¿cuáles son las conclusiones?. ¿Los resultados permiten responder al problemas?	9
7. Código. Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y represntación de los datos.	9

```
library(kableExtra)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:kableExtra':
##
##   group_rows

## The following objects are masked from 'package:stats':
##
##   filter, lag
```

```
## The following objects are masked from 'package:base':
##
## intersect, setdiff, setequal, union
```

1 - Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

El dataset de Titanic: Machine Learning from Disaster se registran los datos de los pasajeros del famoso trasatlántico y se utiliza para predecir los supervivientes. Los datos estan divididos en dos dataset, uno de test y otro entrenamiento, para la creación de modelos de predicción.

2 - Integración y selección de los datos de interés a analizar

Se importan los datos. Primero el dataset train.

```
datostrain <- read.csv("./data/train.csv", stringsAsFactors = F, na.strings = c("NA", ""))
str(datostrain)
```

```
## 'data.frame':    891 obs. of  12 variables:
## $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
## $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
## $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
## $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## $ Sex        : chr  "male" "female" "female" "female" ...
## $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
## $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
## $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
## $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin      : chr  NA "C85" NA "C123" ...
## $ Embarked   : chr  "S" "C" "S" "S" ...
```

Se observa como consta de 891 muestras y 12 variables, entre ellas Survived.

Posteiormente el dataset test.

```
datostest <- read.csv ("./data/test.csv", stringsAsFactors = F, na.strings = c("NA", ""))
str(datostest)
```

```
## 'data.frame':    418 obs. of  11 variables:
## $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
## $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
## $ Name       : chr  "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)" "Myles, Mr. Thomas Francis"
## $ Sex        : chr  "male" "female" "male" "male" ...
## $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
## $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
## $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
## $ Ticket     : chr  "330911" "363272" "240276" "315154" ...
## $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
## $ Cabin      : chr  NA NA NA NA ...
## $ Embarked   : chr  "Q" "S" "Q" "S" ...
```

Se observa como tiene 418 muestra, y 11 variables. La variable Survived no aparece porque es la que se tiene que predecir.

A continuacion, a la hora de fusionar los datos caben dos posibilidades, asignar “NA” a la variable datostest\$Survived o no considerar los datos de survived en train. Se importan, fusionan los datos y se revisa la estructura inicial de los datos.

```
datostest$Survived <- NA
datos <- rbind(datostrain, datostest)
str(datos)
```

```
## 'data.frame': 1309 obs. of 12 variables:
## $ PassengerId: int 1 2 3 4 5 6 7 8 9 10 ...
## $ Survived : int 0 1 1 1 0 0 0 0 1 1 ...
## $ Pclass : int 3 1 3 1 3 3 1 3 3 2 ...
## $ Name : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## $ Sex : chr "male" "female" "female" "female" ...
## $ Age : num 22 38 26 35 35 NA 54 2 27 14 ...
## $ SibSp : int 1 1 0 1 0 0 0 3 0 1 ...
## $ Parch : int 0 0 0 0 0 0 0 1 2 0 ...
## $ Ticket : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin : chr NA "C85" NA "C123" ...
## $ Embarked : chr "S" "C" "S" "S" ...
```

A continuación comprobamos si faltan datos.

Compruebo que no hay valores nulos.

```
# Busco primero qué variables tienen valores perdidos
sapply(datos, function(x) sum(is.na(x)))
```

```
## PassengerId Survived Pclass Name Sex Age
## 0 418 0 0 0 263
## SibSp Parch Ticket Fare Cabin Embarked
## 0 0 0 1 1014 2
```

Podemos observar, que en Survived, salen los 418, que tenemos que predecir, por lo que todos los valores de train están informados.

A continuación se detallan las variables y su tipo inicial, este ultimo, se modificara para su mejor analisis.

```
# datostrain1 <- datostrain[,-2]
# data <- rbind(datostrain1, datostest) # Fusion datasets
data <- datos[,-2]
str(data)
```

```
## 'data.frame': 1309 obs. of 11 variables:
## $ PassengerId: int 1 2 3 4 5 6 7 8 9 10 ...
## $ Pclass : int 3 1 3 1 3 3 1 3 3 2 ...
## $ Name : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## $ Sex : chr "male" "female" "female" "female" ...
## $ Age : num 22 38 26 35 35 NA 54 2 27 14 ...
```

```
## $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
## $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
## $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin      : chr  NA "C85" NA "C123" ...
## $ Embarked   : chr  "S" "C" "S" "S" ...
```

```
tipos <- sapply(data, class)
kable(data.frame(Variables = names(tipos), Tipo_Variable= as.vector(tipos))) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "left")
```

Variables	Tipo_Variable
PassengerId	integer
Pclass	integer
Name	character
Sex	character
Age	numeric
SibSp	integer
Parch	integer
Ticket	character
Fare	numeric
Cabin	character
Embarked	character

De las variables, se convertiran a entero la edad y a factor las siguientes variables Pclass, Sex, Cabin y Embarked.

```
data$Age <- as.integer(data$Age)
data$Pclass <- as.factor(data$Pclass)
data$Sex <- as.factor(data$Sex)
data$Embarked <- as.factor(data$Embarked)
data$Cabin <- as.factor(data$Cabin)

tipos_new <- sapply(data, class)
kable(data.frame(Variables = names(tipos_new), Tipo_Variable= as.vector(tipos_new))) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "left")
```

Variables	Tipo_Variable
PassengerId	integer
Pclass	factor
Name	character
Sex	factor
Age	integer
SibSp	integer
Parch	integer
Ticket	character
Fare	numeric
Cabin	factor
Embarked	factor

Una vez modificadas los tipos de valores se resume que

```
summary(data)
```

```
## PassengerId Pclass Name Sex Age
## Min. : 1 1:323 Length:1309 female:466 Min. : 0.00
## 1st Qu.: 328 2:277 Class :character male :843 1st Qu.:21.00
## Median : 655 3:709 Mode :character Median :28.00
## Mean : 655 Mean :29.86
## 3rd Qu.: 982 3rd Qu.:39.00
## Max. :1309 Max. :80.00
## NA's :263
## SibSp Parch Ticket Fare
## Min. :0.0000 Min. :0.000 Length:1309 Min. : 0.000
## 1st Qu.:0.0000 1st Qu.:0.000 Class :character 1st Qu.: 7.896
## Median :0.0000 Median :0.000 Mode :character Median : 14.454
## Mean :0.4989 Mean :0.385 Mean : 33.295
## 3rd Qu.:1.0000 3rd Qu.:0.000 3rd Qu.: 31.275
## Max. :8.0000 Max. :9.000 Max. :512.329
## NA's :1
## Cabin Embarked
## C23 C25 C27 : 6 C :270
## B57 B59 B63 B66: 5 Q :123
## G6 : 5 S :914
## B96 B98 : 4 NA's: 2
## C22 C26 : 4
## (Other) : 271
## NA's :1014
```

PassengerId: Variable de tipo entero que contiene el id del pasajero, no existen valores nulos o perdidos.

Pclass: Variable de tipo factor con la categoria asignada al pasajero, no existen valores nulos o perdidos.

Name: Variable de tipo texto con el nombre del pasajero, no existen valores nulos o perdidos.

Sex: Variable de tipo factor con el genero del pasajero, no existen valores nulos o perdidos.

Age: Variable de tipo entero que especifica la edad del pasajero, **existen 263 valores nulos**.

SibSp: Variable de tipo entero que especifica el numero de hermanos/esposa abordo, no existen valores nulos o perdidos.

Parch: Variable de tipo entero que especifica el numero de padres/hijos abordo, no existen valores nulos o perdidos.

Ticket: Variable de tipo texto que indica el numero de ticket, no existen valores nulos o perdidos.

Fare: Variable de tipo numero que especifica la tarifa pagada, **existe 1 valor nulo**.

Cabin: Variable de tipo factor donde se especifica la cabina asignada, **existen 1014 valores perdidos**.

Embarked: Variable de tipo factor que indica el puerto de embarque, **existen 2 valores perdidos**.

3 - Limpieza de datos

3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

De las variables existentes a continuación se especifican aquellas que contienen valores perdido o nulos.

Age: *existen 263 valores nulos*.

“Para imputar valores **edad**, (analizando si es mejor imputar mediante la media de los valores, rpart o mice)”

```
sum(is.na(data$Age))      ## Realiza el conteo de valores NA de la variable Age
```

```
## [1] 263
```

Fare: *existe 1 valor nulo.*

Para imputar valores **Fare**

Dado que unicamente hay un valor perdido, es posible imputarlo por la media en base al puerto de embarque "S" y la clase "3"

```
sum(is.na(data$Fare))      ## Realiza el conteo de valores NA de la variable Fare
```

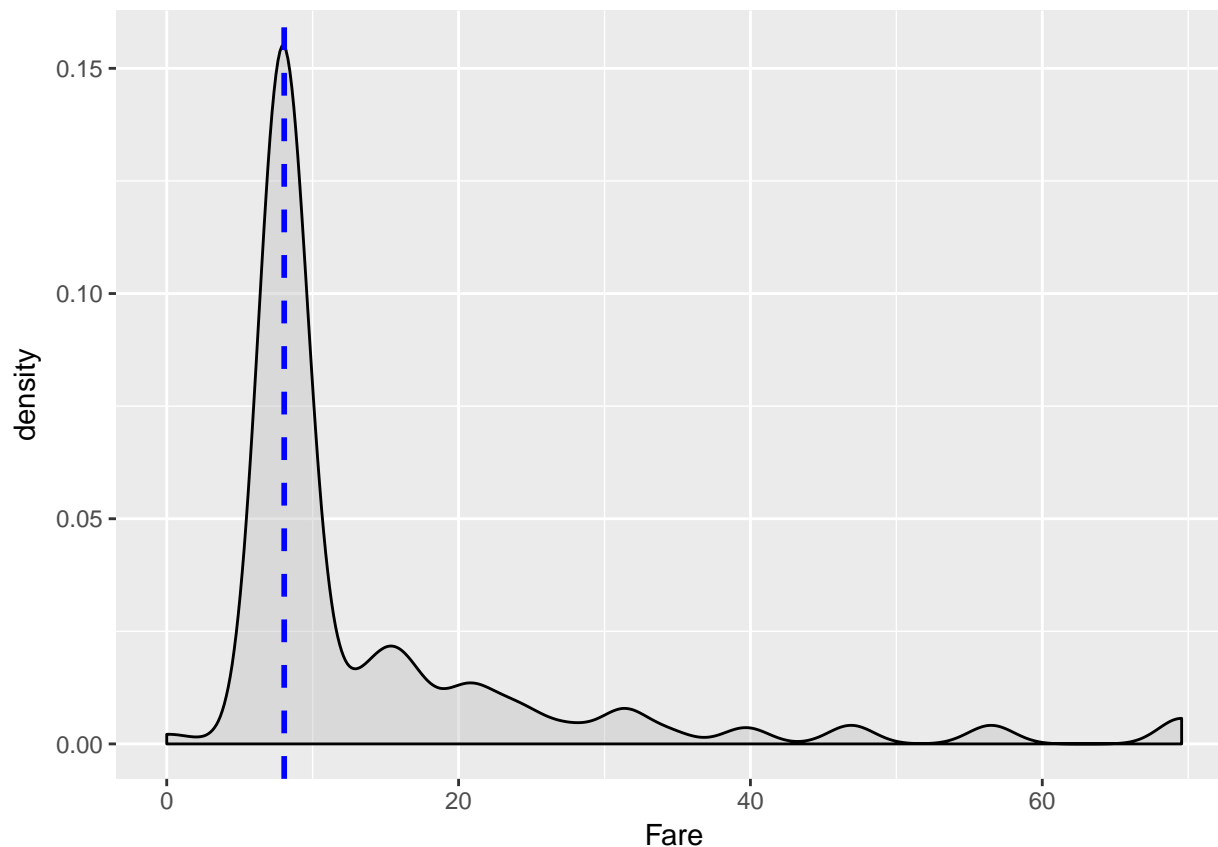
```
## [1] 1
```

```
M_fare<- subset(data,data$Pclass == '3' & data$Embarked == 'S')  
median(M_fare$Fare, na.rm = T)
```

```
## [1] 8.05
```

```
ggplot(M_fare, aes(x = Fare)) +  
  geom_density(fill = 'grey', alpha=0.4) +  
  geom_vline(aes(xintercept=median(Fare, na.rm=T)),  
    colour='blue', linetype='dashed', lwd=1)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```



La tarifa de 80 coincide con la media de los pasajeros de primera clase que embarcaron en C, por lo que se podría imputar este puerto.

```
data$Fare[c(1044)] <- 80
```

Cabin: *existen 1014 valores perdidos.*

Para imputar valores **Cabin**

Esta variable tiene muchos valores perdidos, se podría conseguir predecir la cubierta asignada al pasajero pero es un dato que poco beneficio podría traer ya que se puede realizar el analisis con la combinación entre la tarifa y la clase del pasajero.

```
sum(data$Cabin=="")      ## Realiza el conteo de valores vacios de la variable Cabin
```

```
## [1] NA
```

Embarked: *existen 2 valores perdidos.*

Para imputar valores **Embarked**

Al ser unicamente dos valores perdidos, se podría sustituir los valores por la media, en base a otros pasajeros de la misma clase y puerto de embarque. Los pasajeros han pagado una tarifa de 80 y pertenecian a primera clase.

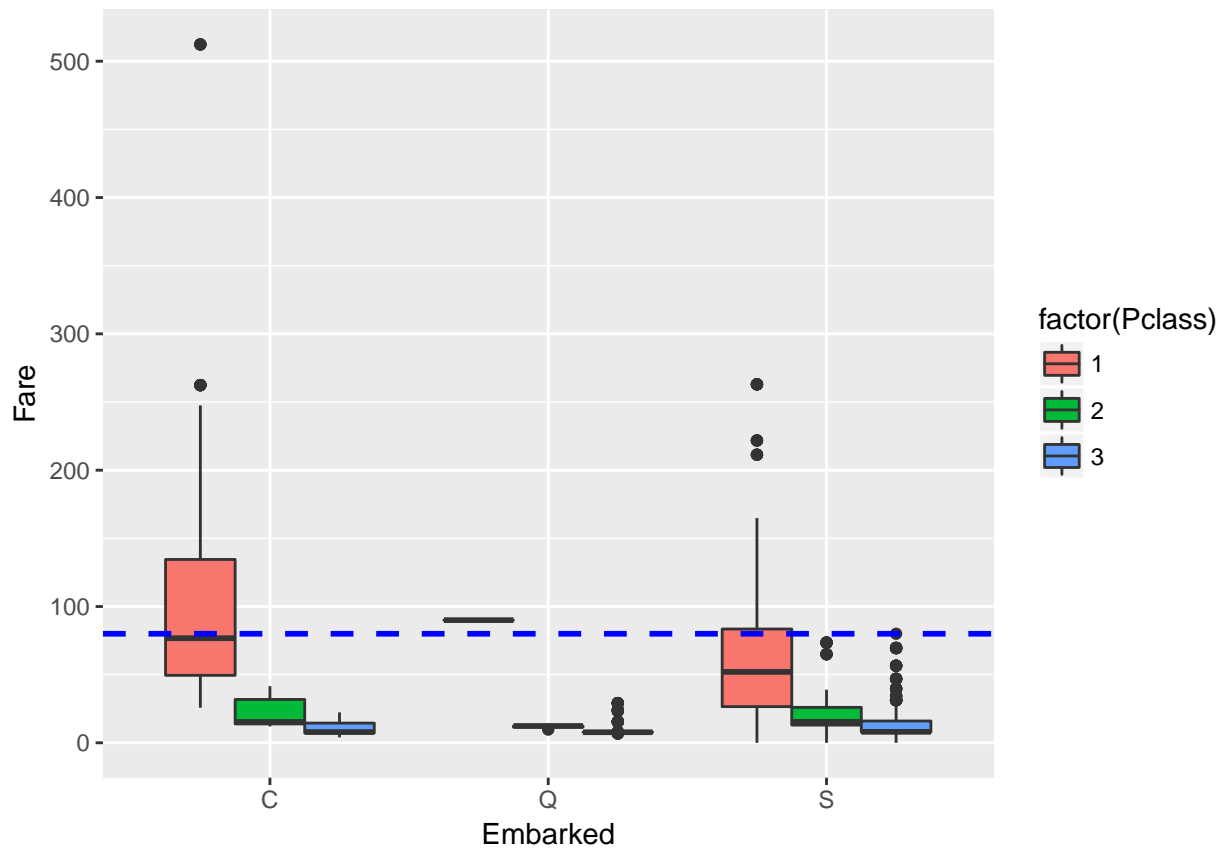
```
data[(data$Embarked==""),]
```

```
##      PassengerId Pclass Name  Sex Age SibSp Parch Ticket Fare Cabin
## NA              NA   <NA> <NA> <NA>  NA    NA    NA   <NA>   NA  <NA>
```

```
## NA.1      NA    <NA> <NA> <NA>  NA    NA    NA    <NA>  NA    <NA>
##      Embarked
## NA      <NA>
## NA.1    <NA>
```

```
embarco <- data %>%
  filter( PassengerId != 62 & PassengerId != 830)

ggplot(embarco, aes(x = Embarked, y = Fare, fill = factor(Pclass))) +
  geom_boxplot() +
  geom_hline(aes(yintercept=80),
    colour='blue', linetype='dashed', lwd=1)
```



La tarifa de 80 coincide con la media de los pasajeros de primera clase que embarcaron en C, por lo que se podría imputar este puerto.

```
data$Embarked[c(62, 830)] <- 'C'
```

3.2. Identificación y tratamiento de valores extremos.

```
##par(mfrow=c(1,3))
##age.bp <- boxplot(data$Age, main= "Edad")
##hist (data$Age, main = "Dist. Edad")
```


4. Análisis de los datos.

4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar)

4.2. Coprobación de la normalidad y homogeneidad de la varianza.

4.3. Aplicación de pruebas estadísticas para comprar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

5. Representación de los reultados a apartir de tablas y gráficas.

6. Resolución del problema. A partir de los resultados obtenidos. ¿cuáles son las conclusiones?. ¿Los resultados permiten responder al problemas?

7. Código. Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y represntación de los datos.